



US009442944B2

(12) **United States Patent**
Dwan et al.

(10) **Patent No.:** **US 9,442,944 B2**
(45) **Date of Patent:** ***Sep. 13, 2016**

(54) **CONTENT ITEM PURGING**

(56) **References Cited**

(71) Applicant: **Dropbox, Inc.**, San Francisco, CA (US)

U.S. PATENT DOCUMENTS

(72) Inventors: **Michael Dwan**, San Francisco, CA (US); **Anthony Grue**, San Francisco, CA (US); **Daniel Kluesing**, Menlo Park, CA (US)

4,973,086 A 11/1990 Donnelly et al.
5,216,592 A 6/1993 Mann et al.
5,235,415 A 8/1993 Bonicel et al.
5,251,294 A 10/1993 Abelow

(Continued)

FOREIGN PATENT DOCUMENTS

(73) Assignee: **Dropbox, Inc.**, San Francisco, CA (US)

EP 2824895 A1 * 1/2015
WO 0063801 A1 10/2000

(Continued)

OTHER PUBLICATIONS

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 245 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **14/077,953**

Zhang, F. et al., Proceedings of the 2009 WRI World Congress on Software Engineering, "Sharing solutions between Java content repository and database" WCSE 2009, 472-5, 2009; ISBN-13: 978-0-7695-3570-8; DOI: 10.1109/WCSE.2009.240; Conference: 2009 WRI World Congress on Software Engineering. WCSE 2009, May 19-21, 2009, Xiamen, China; Sponsor: WRI; Publisher: IEEE, Piscataway, NJ, USA.

(22) Filed: **Nov. 12, 2013**

(Continued)

(65) **Prior Publication Data**

US 2015/0134624 A1 May 14, 2015

Primary Examiner — Thanh-Ha Dang

(74) *Attorney, Agent, or Firm* — Hickman Palermo Becker Bingham LLP

(51) **Int. Cl.**
G06F 17/30 (2006.01)

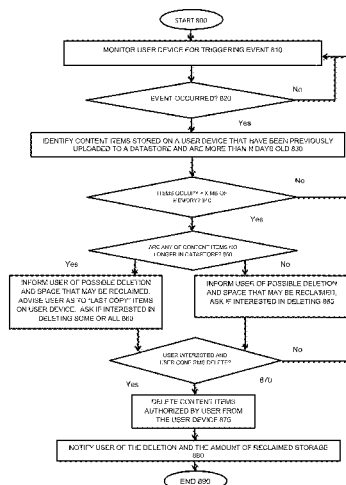
(52) **U.S. Cl.**
CPC ... **G06F 17/30117** (2013.01); **G06F 17/30303** (2013.01)

(58) **Field of Classification Search**
CPC G06F 17/30117; G06F 17/30303;
G06F 11/1451; G06F 12/0253; G06F
17/30348; G06F 17/30566; G06F 17/30867;
G06F 3/0665; H04N 21/4126
USPC 707/692
See application file for complete search history.

(57) **ABSTRACT**

Methods, systems, and computer readable media for content item purging functionality are provided. A contact item purger, such as may be incorporated within a local client application of a content management system, leverages its knowledge as to which items have been uploaded to the content management system, and how long content items have been stored on the user device, to propose items for local deletion and thus reclaiming storage on the user device. A contact item purger may run on one or more devices of a user associated with an account on a content management system upon various triggering events, and may run with or without user interaction, thus maintaining available user device memory capacity at all times.

26 Claims, 9 Drawing Sheets



(56)

References Cited**U.S. PATENT DOCUMENTS**

5,307,266 A 4/1994 Hayashi et al.
 5,412,566 A 5/1995 Sawa
 5,418,713 A 5/1995 Allen
 5,483,466 A 1/1996 Kawahara et al.
 5,513,356 A 4/1996 Takahashi et al.
 5,541,394 A 7/1996 Kouchi et al.
 5,575,573 A 11/1996 Ito et al.
 5,666,585 A 9/1997 Nagira et al.
 5,752,021 A 5/1998 Nakatsuyama et al.
 5,787,413 A 7/1998 Kauffman et al.
 5,787,446 A 7/1998 Dang et al.
 5,794,217 A 8/1998 Allen
 5,842,213 A 11/1998 Odom et al.
 5,870,716 A 2/1999 Sugiyama et al.
 5,873,086 A 2/1999 Fujii et al.
 5,896,358 A 4/1999 Endoh et al.
 5,905,981 A 5/1999 Lawler
 5,935,243 A 8/1999 Hasebe et al.
 6,049,777 A 4/2000 Sheena et al.
 6,269,382 B1 7/2001 Cabrera et al.
 7,044,365 B2 5/2006 Witherspoon
 7,054,887 B2 5/2006 Kozina
 7,117,519 B1 10/2006 Anderson et al.
 7,171,480 B2 1/2007 Chatani
 7,343,568 B2 3/2008 Jiang et al.
 7,386,476 B1 6/2008 Shavanadan et al.
 7,426,533 B2 9/2008 Malone et al.
 7,444,588 B2 10/2008 Hill et al.
 7,519,671 B2 4/2009 Ho
 7,523,132 B2 4/2009 Altounian et al.
 7,584,225 B2 9/2009 Jiang et al.
 7,698,516 B2 4/2010 Todd et al.
 7,865,497 B1 1/2011 Pillai
 7,917,417 B2 3/2011 Dion
 7,958,196 B2 6/2011 Malone et al.
 7,996,172 B2 8/2011 Bauer et al.
 8,005,874 B2 8/2011 Altounian et al.
 8,032,113 B2 10/2011 Bailey et al.
 8,065,385 B2 11/2011 Altounian et al.
 8,121,902 B1 2/2012 Desjardins et al.
 8,255,545 B1* 8/2012 Schmidt H04L 67/1095
 709/227
 8,261,205 B2 9/2012 Lee
 2002/0174200 A1 11/2002 Kozina
 2003/0033254 A1* 2/2003 Tanaka H04L 63/04
 705/57
 2004/0039887 A1* 2/2004 Gautney H04L 67/125
 711/159
 2007/0091836 A1* 4/2007 Opreescu-Surcobe
 et al. 370/318
 2007/0120959 A1* 5/2007 Wu et al. 348/14.02
 2008/0250024 A1* 10/2008 Kvm et al. 707/10
 2008/0311961 A1* 12/2008 Cotevino et al. 455/574
 2009/0254966 A1* 10/2009 Josephs H04N 7/17318
 725/140
 2010/0042801 A1* 2/2010 Lee G06F 9/5016
 711/171
 2010/0070448 A1 3/2010 Omoigui

2010/0306175 A1* 12/2010 Johnson et al. 707/663
 2011/0258333 A1* 10/2011 Pomerantz H04L 63/0428
 709/229
 2012/0005041 A1* 1/2012 Mehta G06Q 30/0601
 705/27.1
 2012/0005307 A1 1/2012 Das et al.
 2012/0047108 A1* 2/2012 Mandel G06F 3/0607
 707/639
 2012/0117406 A1* 5/2012 Eun H04W 52/0216
 713/323
 2012/0131125 A1 5/2012 Seidel et al.
 2012/0136727 A1* 5/2012 Guioguo, Jr. G06Q 30/0241
 705/14.69
 2012/0191716 A1 7/2012 Omoigui
 2012/0215878 A1 8/2012 Kidron
 2012/0216296 A1 8/2012 Kidron
 2012/0221382 A1 8/2012 Kidron
 2012/0221559 A1 8/2012 Kidron
 2012/0221951 A1 8/2012 Kidron
 2012/0222050 A1 8/2012 Kidron
 2012/0222125 A1 8/2012 Kidron
 2012/0222133 A1 8/2012 Kidron
 2012/0254340 A1* 10/2012 Velumylum et al. 709/213
 2013/0115981 A1* 5/2013 Toksvig et al. 455/466
 2013/0210493 A1* 8/2013 Tal et al. 455/566
 2013/0311742 A1* 11/2013 Ding et al. 711/170

FOREIGN PATENT DOCUMENTS

WO WO00/63801 A1 10/2000
 WO WO0063801 A1 10/2000

OTHER PUBLICATIONS

Wu, C. et al., Personal and Ubiquitous Computing, "Point-of-capture archiving and editing of personal experiences from a mobile device" v 11, n 4, p. 235-249, Apr. 2007, Memory and Sharing of Experiences; ISSN: 16174909; DOI: 10.1007/s00779-006-0082-7; Publisher: Springer London Ltd.
 Zhang et al., "Sharing Solutions Between Java Content Repository and Database", WCSE, dated 2009, WRI World Congress on Software Engineering, 4 pages.
 Wu, C. et al., "Point of-Capture Archiving and Editing of Personal Experiences From a Mobile Device", dated Apr. 2007, Memory Sharing of Experiences, 50 pages.
 Zhang et al., "Sharing Solutions Between Java Content Repository and Database", WCSE dated May 2009.
 Wu et al., "Personal and Ubiquitous Computing, Point of Capture Archiving and Editing of Personal Experiences from a Mobile Device", dated Apr. 2007.
 U.S. Appl. No. 14/085,760, filed Nov. 20, 2013, Final Office Action, Jun. 13, 2014.
 U.S. Appl. No. 14/085,760, filed Nov. 20, 2013, Advisory Action, Aug. 27, 2014.
 U.S. Appl. No. 14/085,760, filed Nov. 20, 2013, Office Action, Jan. 30, 2014.
 U.S. Appl. No. 14/085,760, filed Nov. 20, 2013, Notice of Allowance, Sep. 22, 2014.

* cited by examiner

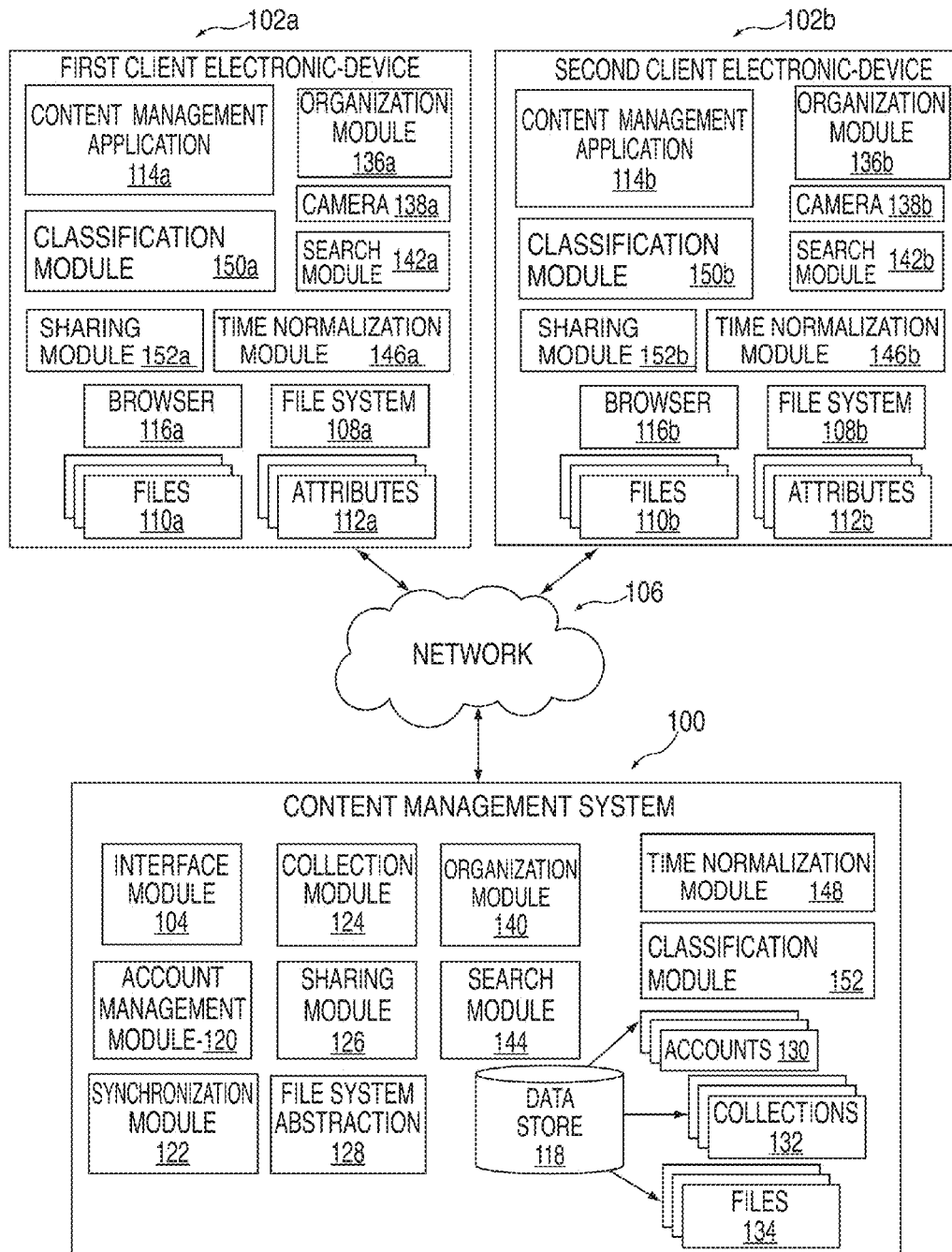


FIG. 1

Fig. 2

Camera Roll Housekeeping

We found 55 photos and 3
Videos that were already
uploaded to [*ServiceSystem*]

Would you like to delete these
Files from your device and free
185MB of space?

No Thanks Delete

Fig. 3



Fig. 4

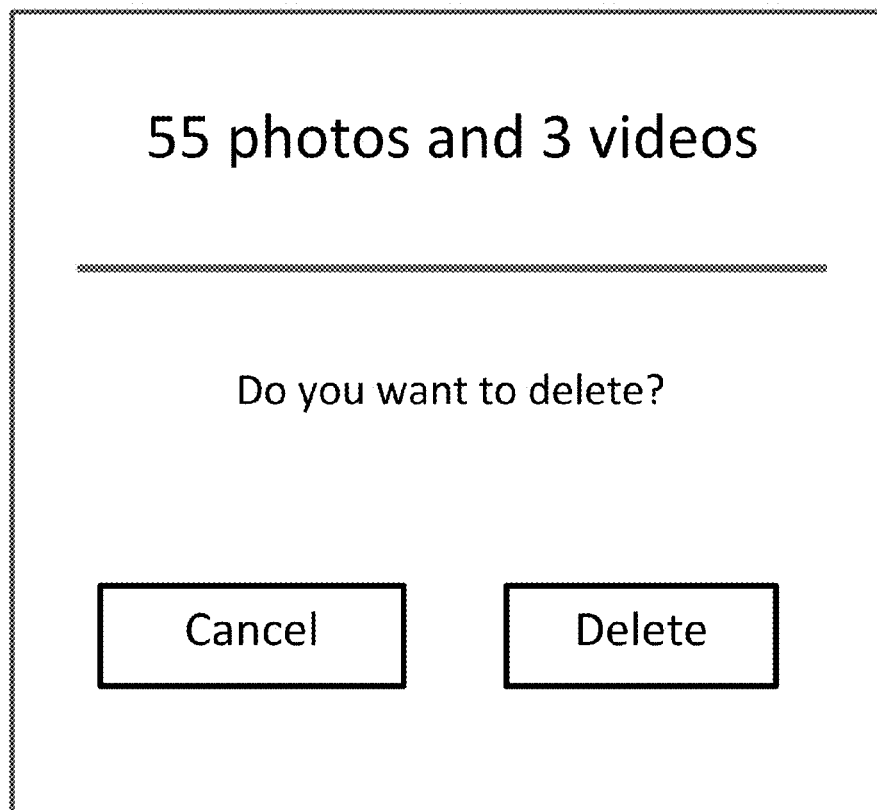


Fig. 5

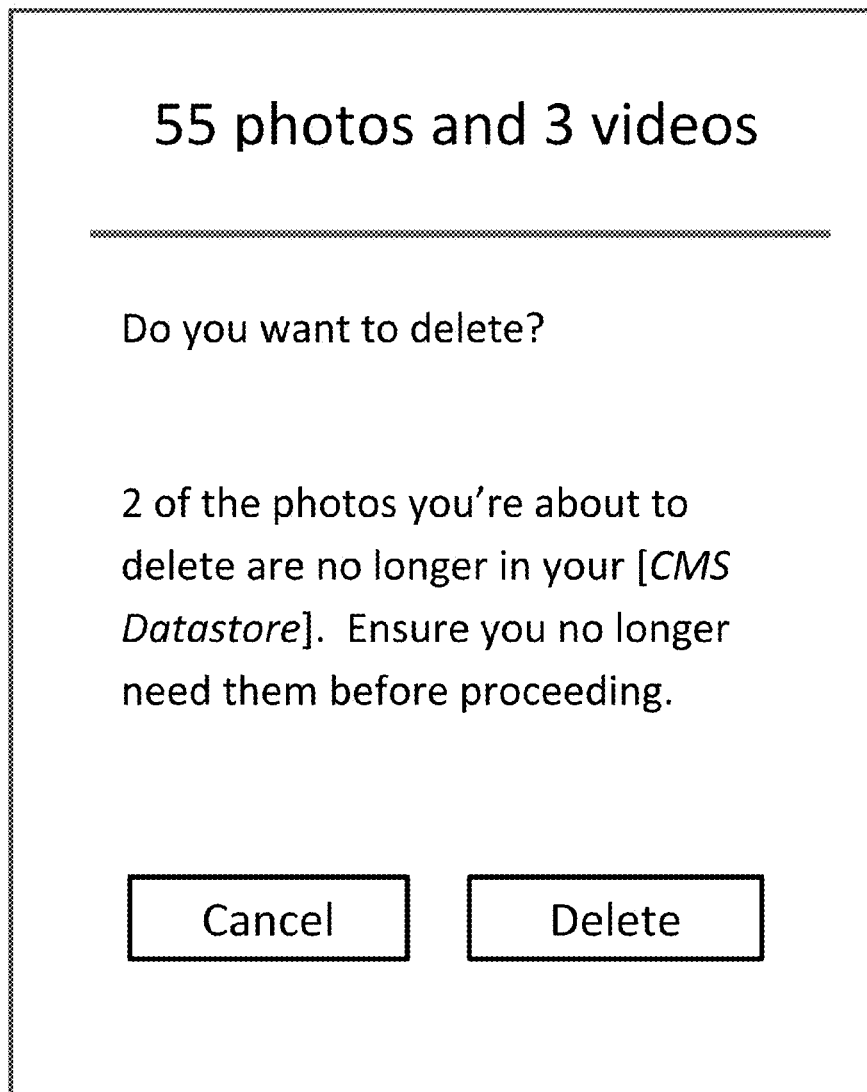


Fig. 6

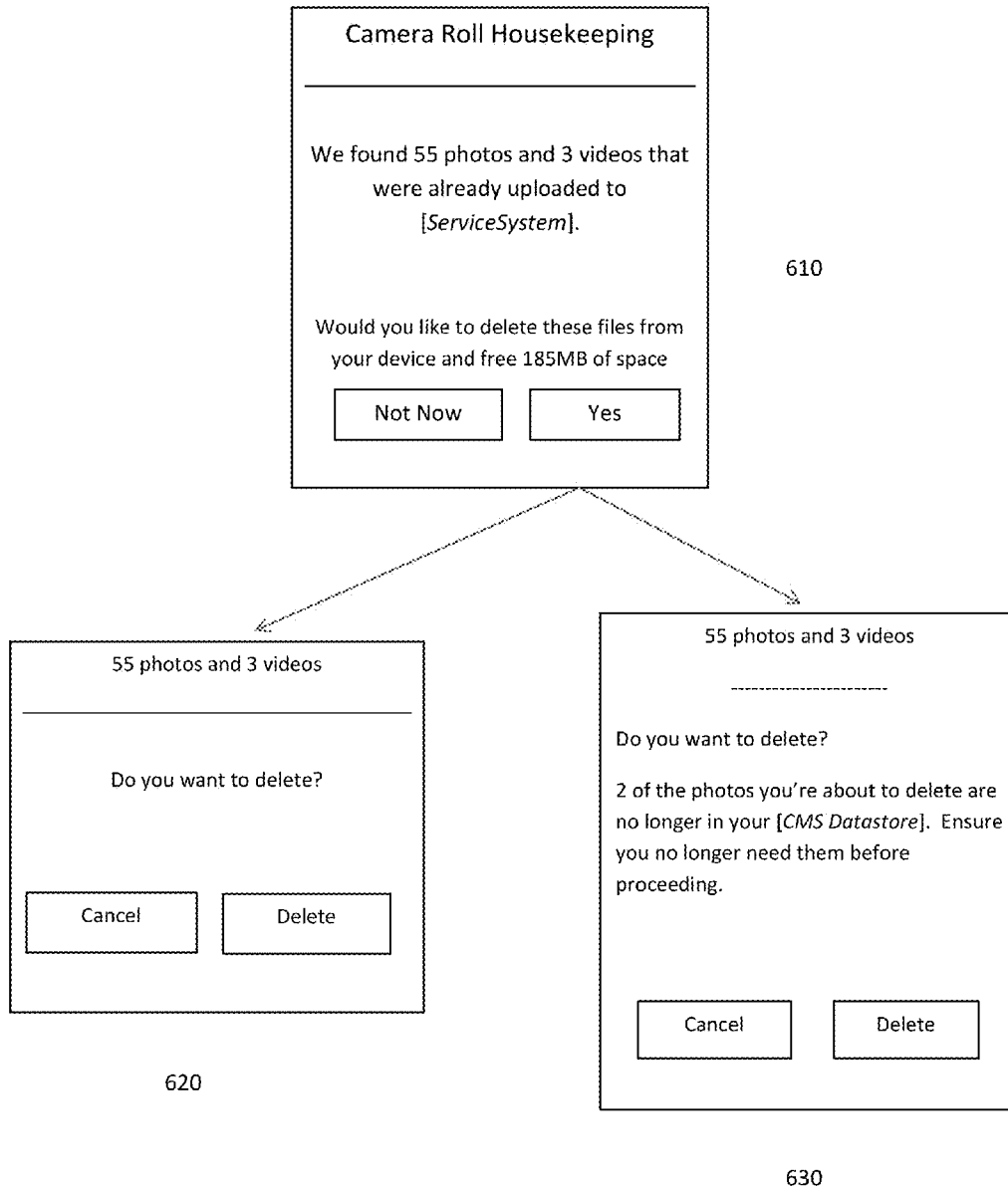


FIG. 7

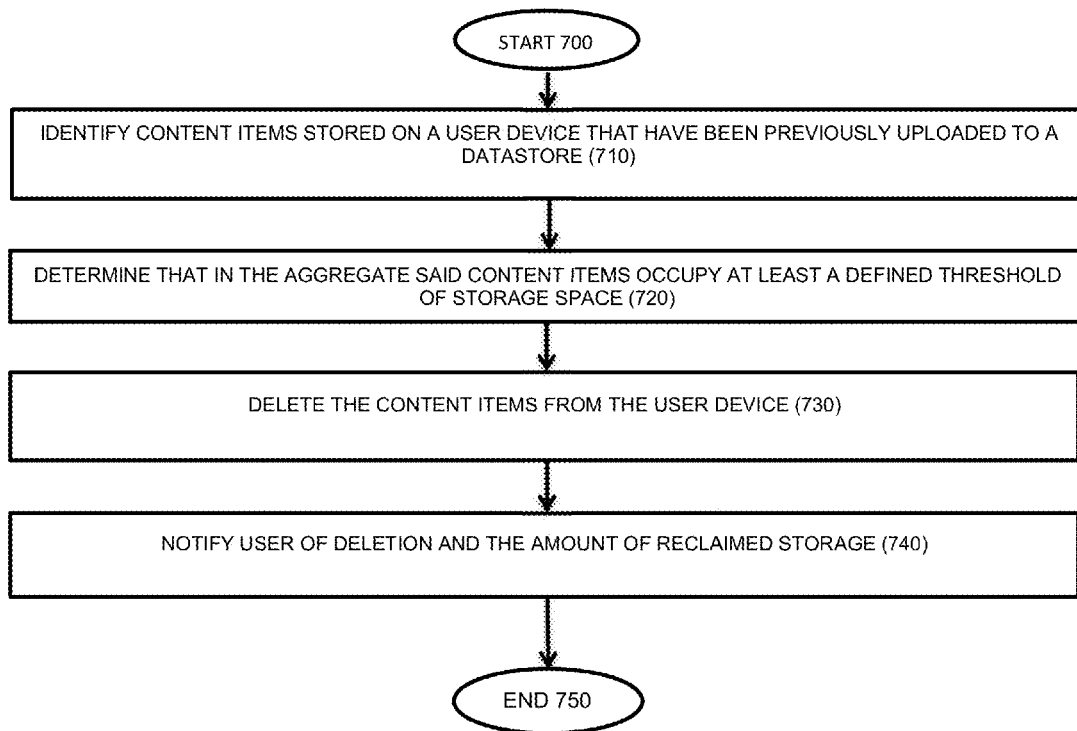
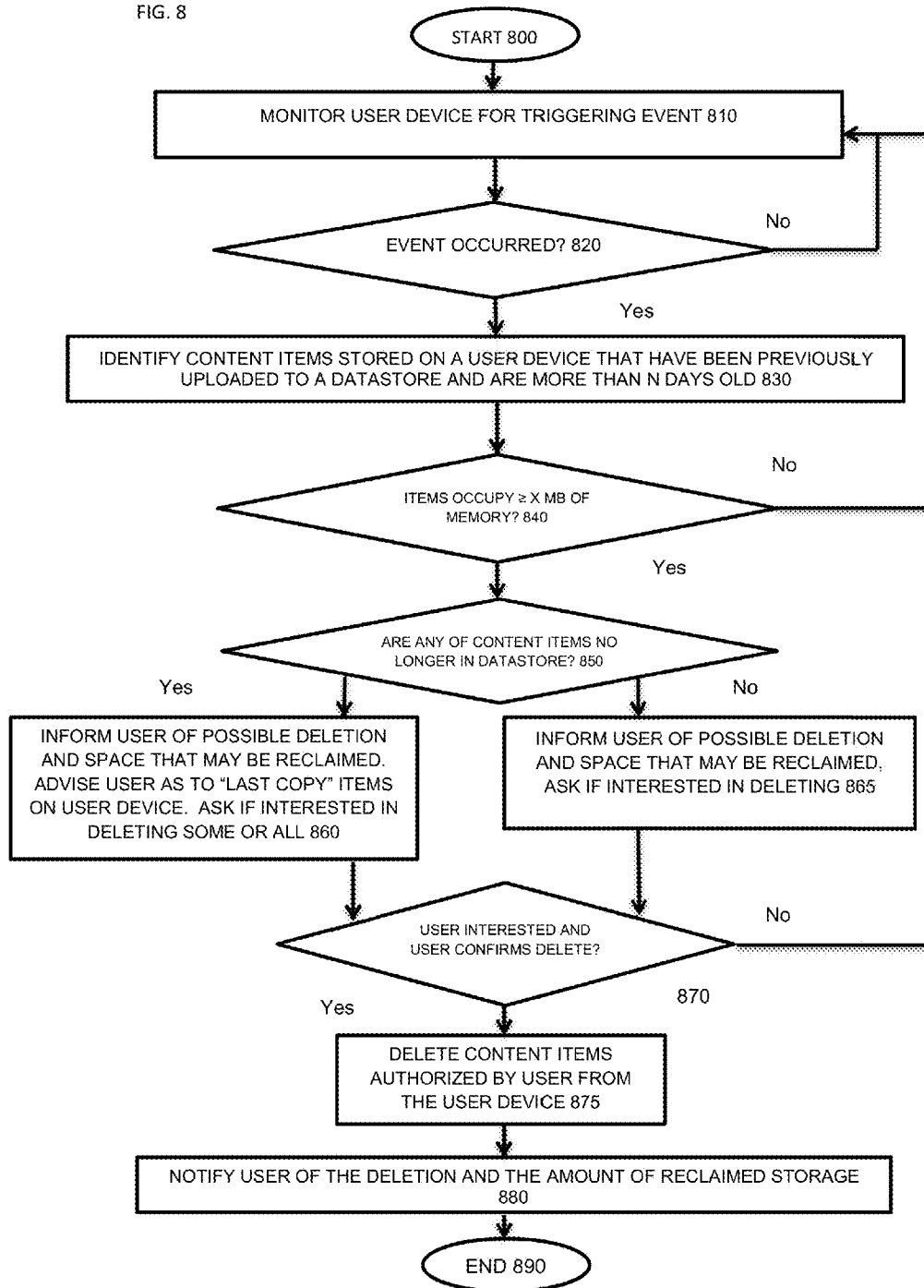


FIG. 8



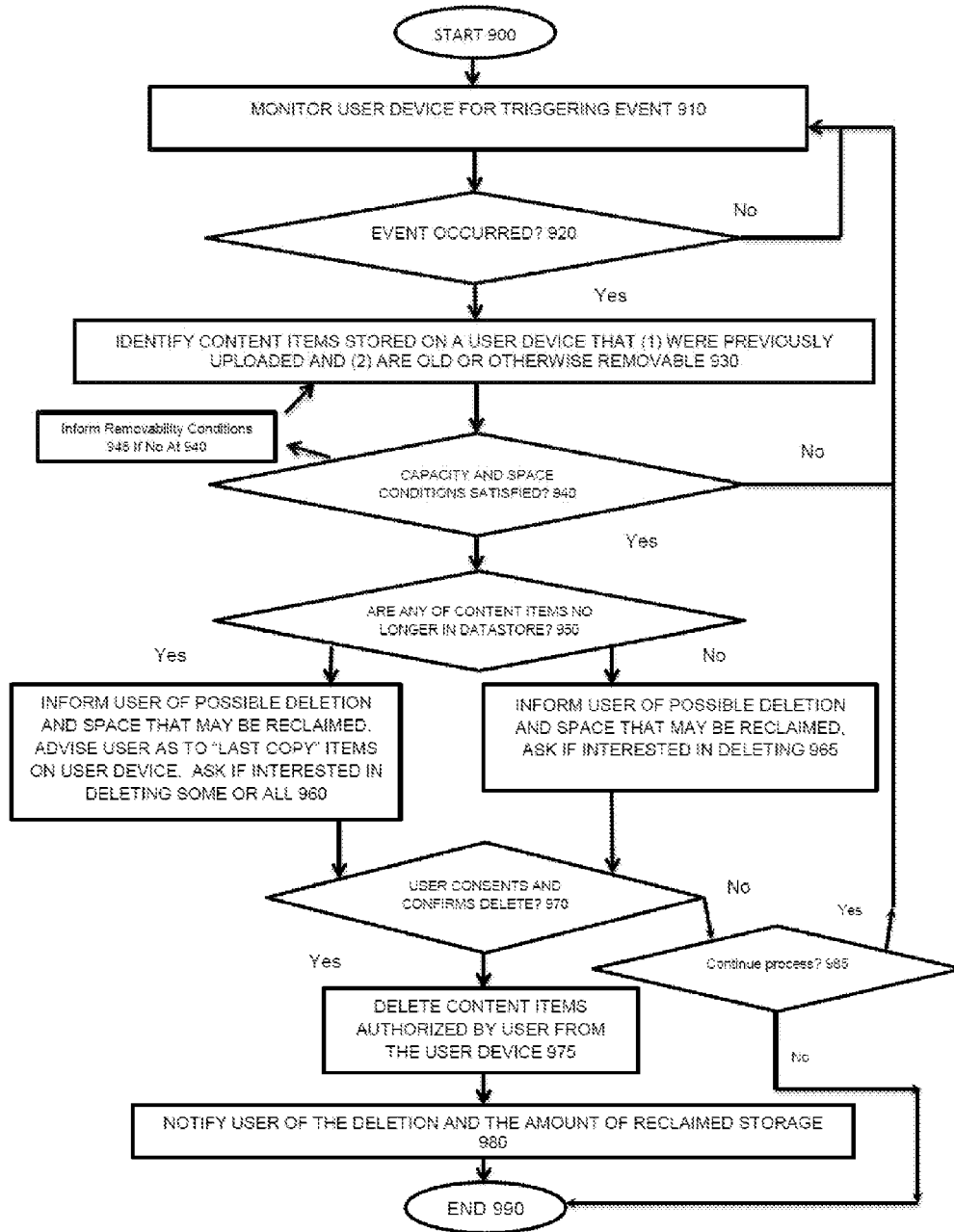


FIG. 9

1

CONTENT ITEM PURGING**FIELD OF THE INVENTION**

Various embodiments relate generally to content item purging.

BACKGROUND

Users of various Internet accessible data storage and management services, such as, for example, content management systems, often upload large numbers of content items from various user devices to a datastore on such a system. For example, users of various content management systems may take photographs with a smartphone or similar device, and then upload them from their device to one or more accounts on one or more content management systems. Or, similarly, users may acquire various audio files on a device (e.g., by recording with the device, or saving from an email) and subsequently upload them to a content management system. If this is done often, the user may soon realize that his or her memory capacity on the user device is at or very near full capacity. In such an event, a user must delete some content. Obviously content that she has already placed in a remote datastore may be the best items to locally delete. But the user does not remember such details.

Accordingly, there is a need to leverage the fact that a content management system may readily identify which content items on a user device have already been uploaded to one or more datastores maintained by the content management system, and thus periodically inform a user that space could be saved on his or her user device by purging such already saved content from the device.

SUMMARY

Embodiments are provided for purging content items from a user device. Methods, systems, and computer readable media for a contact item purging functionality are provided. A contact item purger, such as may be incorporated within a local client application of a content management system leverages its knowledge as to what has been uploaded to the content management system, and when, to propose content items that may safely be deleted locally and thus reclaim storage on the user device. A contact item purger may run on one or more devices of a user associated with an account on a content management system, and may continually manage potential items to delete, thus maintaining the user device memory available at all times.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other aspects and advantages of the invention will become more apparent upon consideration of the following detailed description, taken in conjunction with accompanying drawings, in which like reference characters refer to like parts throughout, and in which:

FIG. 1 is an exemplary system in which content purge functionality may be provided in accordance with some embodiments of the invention;

FIGS. 2-6 depict exemplary user interfaces for a content item purging functionality in accordance with some embodiments of the invention;

FIG. 7 is an exemplary process flow diagram for an illustrative basic content item purger in accordance with some embodiments of the invention;

2

FIG. 8 is an exemplary process flow diagram for a more detailed illustrative content item purger in accordance with some embodiments of the invention; and

FIG. 9 is an exemplary process flow diagram for still another detailed illustrative content item purger in accordance with some embodiments of the invention.

DETAILED DESCRIPTION OF THE DISCLOSURE

Methods, systems, and computer readable media for content purging/space saving functionality on user devices are provided. Methods, systems, and computer readable media for a contact item purging functionality are provided. A contact item purger, such as may be incorporated within a local client application of a content management system leverages its knowledge as to what has been uploaded to the content management system, and when, to propose content items that may safely be deleted locally and thus reclaim storage on the user device. A contact item purger may run on one or more devices of a user associated with an account on a content management system, and may continually manage potential items to delete, thus maintaining the user device memory available at all times.

As noted, for purposes of description and simplicity, methods, systems and computer readable mediums will be described for a content storage and management service, and in particular, for a contact importer and processing technology for use in connection with such a content storage and management service. However, the terms “content storage service” and “content management system” are used herein to refer broadly to a variety of storage providers and management service providers as well as systems, services platforms and entities handling a wide variety of types of content, files, portions of files, and/or other types of data. Those with skill in the art will recognize that the methods, systems, and mediums described may be used for a variety of storage providers/services and types of content, files, portions of files, and/or other types of data.

FIG. 1 depicts an exemplary system for which a content item purging functionality may be provided in accordance with some embodiments of the invention. Elements in FIG. 1, including, but not limited to, first client electronic device 102a, second client electronic device 102b, and content management system 100, may communicate by sending and/or receiving data over network 106. Network 106 may be any network, combination of networks, or network devices that may carry data communication. For example, network 106 may be any one or any combination of LAN (local area network), WAN (wide area network), telephone network, wireless network, point-to point network, star network, token ring network, hub network, or any other configuration.

Network 106 may support any number of protocols, including but not limited to TCP/IP (Transfer Control Protocol and Internet Protocol), HTTP (Hypertext Transfer Protocol), WAP (wireless application protocol), etc. For example, first client electronic device 102a and second client electronic device 102b (collectively 102) may communicate with content management system 100 using TCP/IP, and, at a higher level, use browser 116 to communicate with a web server (not shown) at content management system 100 using HTTP. Examples of implementations of browser 116, include, but are not limited to, Google Inc. Chrome™ browser, Microsoft Internet Explorer®, Apple Safari®, Mozilla Firefox, and Opera Software Opera.

A variety of client electronic devices **102** may communicate with content management system **100**, including, but not limited to, desktop computers, mobile computers, mobile communication devices (e.g., mobile phones, smart phones, tablets), televisions, set-top boxes, and/or any other network enabled device. Although two client electronic devices **102a** and **102b** are illustrated for description purposes, those with skill in the art will recognize that any number of devices may be used and supported by content management system **100**. Client electronic devices **102** may be used to create, access, modify, and manage files **110a** and **110b** (collectively **110**) (e.g. files, file segments, images, etc.) stored locally within file system **108a** and **108b** (collectively **108**) on client electronic device **102** and/or stored remotely with content management system **100** (e.g., within data store **118**). For example, client electronic device **102a** may access file **110b** stored remotely with data store **118** of content management system **100** and may or may not store file **110b** locally within file system **108a** on client electronic device **102a**. Continuing with the example, client electronic device **102a** may temporarily store file **110b** within a cache (not shown) locally within client electronic device **102a**, make revisions to file **110b**, and the revisions to file **110b** may be communicated and stored in data store **118** of content management system **100**. Optionally, a local copy of the file **110a** may be stored on client electronic device **102a**.

Client devices **102** may capture, record, and/or store content items, such as image files **110**. Client devices **102** may have a camera **138** (e.g., **138a** and **138b**) to capture and record digital images and/or videos. For example, camera **138** may capture and record images and store metadata with the images. Metadata may include, but is not limited to, the following: creation time timestamp, geo-location, orientation, rotation, title, and/or any other attributes or data relevant to the captured image.

Metadata values may be stored as attribute **112** name-value pairs, tag-value pairs, and/or any other method to associate the metadata with the file and easily identify the type of metadata. In some embodiments, attributes **112** may be tag-value pairs defined by a particular standard, including, but not limited to, Exchangeable Image File Format (Exif), JPEG File Interchange Format (Jfif), and/or any other standard.

A time normalization module **146** (e.g., **146a** and **146b**) may be used to normalize dates and times stored with a content item. An example of time normalization is provided in U.S. Utility patent application Ser. No. 13/888,118, entitled "Date and Time Handling," filed on May 6, 2013, and herein incorporated by reference in its entirety. The time normalization module **146**, counterpart time normalization module **148**, and/or any combination thereof may be used to normalize dates and times stored for content items. The normalized times and dates may be used to sort, group, perform comparisons, perform basic math, and/or cluster content items.

An organization module **136** (e.g., **136a** and **136b**) may be used to organize content items (e.g., image files) into clusters, organize content items to provide samplings of content items for display within user interfaces, and/or retrieve organized content items for presentation. An example of organization is described in U.S. Utility patent application Ser. No. 13/888,186, entitled "Presentation and Organization of Content," filed on May 6, 2013, and herein incorporated by reference in its entirety.

The organization module **136** may utilize any clustering algorithm. The organization module **136** may be used to identify similar images for clusters in order to organize

content items for presentation within user interfaces on devices **102** and content management system **100**. Similarity rules may be defined to create one or more numeric representations embodying information on similarities between each of the content items in accordance with the similarity rules. The organization module **136** may use the numeric representation as a reference for similarity between content items in order to cluster the content items.

In some embodiments, content items may be organized into clusters to aid with retrieval of similar content items in response to search requests. For example, organization module **136a** may identify first and second images are similar and may be group the images together in a cluster. Organization module **136a** may process image files to determine clusters independently or in conjunction with counterpart organization module (e.g., **140** and/or **136b**). In other embodiments, organization module **136a** may only provide clusters identified with counterpart organization modules (e.g., **140** and/or **136b**) for presentation. Continuing with the example, processing of image files to determine clusters may be an iterative process that is executed upon receipt of new content items and/or new similarity rules.

In some embodiments, a search module **142** on client device **102** is provided with counterpart search module **144** on content management system **144** to support search for content items. A search request may be received by search module **142** and/or **144** that requests a content item. In some embodiments, the search may be handled by searching metadata and/or attributes assigned to content items during the provision of management services. For example, cluster markers stored with images may be used to find images by date. In particular, cluster markers may indicate an approximate time or average time for the images stored with the cluster marker in some embodiments, and the marker may be used to speed the search and/or return the search results with the contents of the cluster with particular cluster markers.

Files **110** managed by content management system **100** may be stored locally within file system **108** of respective devices **102** and/or stored remotely within data store **118** of content management system **100** (e.g., files **134** in data store **118**). Content management system **100** may provide synchronization of files managed by content management system **100**. Attributes **112a** and **112b** (collectively **112**) or other metadata may be stored with files **110**. For example, a particular attribute may be stored with the file to track files locally stored on client devices **102** that are managed and/or synchronized by content management system **100**. In some embodiments, attributes **112** may be implemented using extended attributes, resource forks, or any other implementation that allows for storing metadata with a file that is not interpreted by a file system. In particular, an attribute **112a** and **112b** may be a content identifier for a file. For example, the content identifier may be a unique or nearly unique identifier (e.g., number or string) that identifies the file.

By storing a content identifier with the file, a file may be tracked. For example, if a user moves the file to another location within the file system **108** hierarchy and/or modifies the file, then the file may still be identified within the local file system **108** of a client device **102**. Any changes or modifications to the file identified with the content identifier may be uploaded or provided for synchronization and/or version control services provided by the content management system **100**.

A stand-alone content management application **114a** and **114b** (collectively **114**), client application, and/or third-party application may be implemented to provide a user interface for a user to interact with content management system **100**.

5

Content management application **114** may expose the functionality provided with content management interface **104** and accessible modules for device **102**. Web browser **116a** and **116b** (collectively **116**) may be used to display a web page front end for a client application that may provide content management **100** functionality exposed/provided with content management interface **104**. First client electronic device **102a** may include classification module **150a**. Second client electronic device **102b** may include classification module **150b**.

Content management system **100** may allow a user with an authenticated account to store content, as well as perform management tasks, such as retrieve, modify, browse, synchronize, and/or share content with other accounts. Various embodiments of content management system **100** may have elements, including, but not limited to, content management interface module **104**, account management module **120**, synchronization module **122**, collections module **124**, sharing module **126**, file system abstraction **128**, data store **118**, classification module **152**, and organization module **140**. The content management service interface module **104** may expose the server-side or back end functionality/capabilities of content management system **100**. For example, a counterpart user interface (e.g., stand-alone application, client application, etc.) on client electronic devices **102** may be implemented using content management service interface **104** to allow a user to perform functions offered by modules of content management system **100**. In particular, content management system **100** may have an organization module **140** for identifying similar content items for clusters and samples of content items for presentation within user interfaces.

The user interface offered on client electronic device **102** may be used to create an account for a user and authenticate a user to use an account using account management module **120**. The account management module **120** of the content management service may provide the functionality for authenticating use of an account by a user and/or a client electronic device **102** with username/password, device identifiers, and/or any other authentication method. Account information **130** may be maintained in data store **118** for accounts. Account information may include, but is not limited to, personal information (e.g., an email address or username), account management information (e.g., account type, such as “free” or “paid”), usage information, (e.g., file edit history), maximum storage space authorized, storage space used, content storage locations, security settings, personal configuration settings, content sharing data, etc. An amount of content management may be reserved, allotted, allocated, stored, and/or may be accessed with an authenticated account. The account may be used to access files **110** within data store **118** for the account and/or files **110** made accessible to the account that are shared from another account. Account module **120** may interact with any number of other modules of content management system **100**.

An account may be used to store content, such as documents, text files, audio files, video files, etc., from one or more client devices **102** authorized on the account. The content may also include folders of various types with different behaviors, or other mechanisms of grouping content items together. For example, an account may include a public folder that is accessible to any user. The public folder may be assigned a web-accessible address. A link to the web-accessible address may be used to access the contents of the public folder. In another example, an account may include a photos folder that is intended for photos and that provides specific attributes and actions tailored for photos; an audio folder that provides the ability to play back audio

6

files and perform other audio related actions; or other special purpose folders. An account may also include shared folders or group folders that are linked with and available to multiple user accounts. The permissions for multiple users may be different for a shared folder.

Content items (e.g., files **110**) may be stored in data store **118**. Data store **118** may be a storage device, multiple storage devices, or a server. Alternatively, data store **118** may be cloud storage provider or network storage accessible via one or more communications networks. Content management system **100** may hide the complexity and details from client devices **102** by using a file system abstraction **128** (e.g., a file system database abstraction layer) so that client devices **102** do not need to know exactly where the content items are being stored by the content management system **100**. Embodiments may store the content items in the same folder hierarchy as they appear on client device **102**. Alternatively, content management system **100** may store the content items in various orders, arrangements, and/or hierarchies. Content management system **100** may store the content items in a network accessible storage (SAN) device, in a redundant array of inexpensive disks (RAID), etc. Content management system **100** may store content items using one or more partition types, such as FAT, FAT32, NTFS, EXT2, EXT3, EXT4, ReiserFS, BTRFS, and so forth.

Data store **118** may also store metadata describing content items, content item types, and the relationship of content items to various accounts, folders, collections, or groups. The metadata for a content item may be stored as part of the content item and/or may be stored separately. Metadata may be store in an object-oriented database, a relational database, a file system, or any other collection of data. In one variation, each content item stored in data store **118** may be assigned a system-wide unique identifier.

Data store **118** may decrease the amount of storage space required by identifying duplicate files or duplicate chunks of files. Instead of storing multiple copies, data store **118** may store a single copy of a file **134** and then use a pointer or other mechanism to link the duplicates to the single copy. Similarly, data store **118** may store files **134** more efficiently, as well as provide the ability to undo operations, by using a file version control that tracks changes to files, different versions of files (including diverging version trees), and a change history. The change history may include a set of changes that, when applied to the original file version, produce the changed file version.

Content management system **100** may be configured to support automatic synchronization of content from one or more client devices **102**. The synchronization may be platform independent. That is, the content may be synchronized across multiple client devices **102** of varying type, capabilities, operating systems, etc. For example, client device **102a** may include client software, which synchronizes, via a synchronization module **122** at content management system **100**, content in client device **102** file system **108** with the content in an associated user account. In some cases, the client software may synchronize any changes to content in a designated folder and its sub-folders, such as new, deleted, modified, copied, or moved files or folders. In one example of client software that integrates with an existing content management application, a user may manipulate content directly in a local folder, while a background process monitors the local folder for changes and synchronizes those changes to content management system **100**. In some embodiments, a background process may identify content that has been updated at content management system **100**

and synchronize those changes to the local folder. The client software may provide notifications of synchronization operations, and may provide indications of content statuses directly within the content management application. Sometimes client device **102** may not have a network connection available. In this scenario, the client software may monitor the linked folder for file changes and queue those changes for later synchronization to content management system **100** when a network connection is available. Similarly, a user may manually stop or pause synchronization with content management system **100**.

A user may also view or manipulate content via a web interface generated and served by user interface module **104**. For example, the user may navigate in a web browser to a web address provided by content management system **100**. Changes or updates to content in the data store **118** made through the web interface, such as uploading a new version of a file, may be propagated back to other client devices **102** associated with the user's account. For example, multiple client devices **102**, each with their own client software, may be associated with a single account and files in the account may be synchronized between each of the multiple client devices **102**.

Content management system **100** may include sharing module **126** for managing sharing content and/or collections of content publicly or privately. Sharing module **126** may manage sharing independently or in conjunction with counterpart sharing module (e.g., **152a** and **152b**). Sharing content publicly may include making the content item and/or the collection accessible from any computing device in network communication with content management system **100**. Sharing content privately may include linking a content item and/or a collection in data store **118** with two or more user accounts so that each user account has access to the content item. The sharing may be performed in a platform independent manner. That is, the content may be shared across multiple client devices **102** of varying type, capabilities, operating systems, etc. The content may also be shared across varying types of user accounts. In particular, the sharing module **126** may be used with the collections module **124** to allow sharing of a virtual collection with another user or user account. A virtual collection may be a grouping of content identifiers that may be stored in various locations within file system of client device **102** and/or stored remotely at content management system **100**.

The virtual collection for an account with a file storage service is a grouping of one or more identifiers for content items (e.g., identifying content items in storage). An example of virtual collections is provided in U.S. patent application Ser. No. 14/054,103, entitled "Systems and Methods for Presenting Content Items in a Collections View," filed on Oct. 15, 2013, which claims the benefit of U.S. Provisional Patent Application No. 61/750,791, entitled "Presenting Content Items in a Collections View," filed on Jan. 9, 2013, both of which are incorporated herein by reference in their entirety. The virtual collection is created with the collection module **124** by selecting from existing content items stored and/or managed by the file storage service and associating the existing content items within data storage (e.g., associating storage locations, content identifiers, or addresses of stored content items) with the virtual collection. By associating existing content items with the virtual collection, a content item may be designated as part of the virtual collection without having to store (e.g., copy and paste the content item file to a directory) the content item in another location within data storage in order to place the content item in the collection.

In some embodiments, content management system **100** may be configured to maintain a content directory or a database table/entity for content items where each entry or row identifies the location of each content item in data store **118**. In some embodiments, a unique or a nearly unique content identifier may be stored for each content item stored in the data store **118**.

Metadata may be stored for each content item. For example, metadata may include a content path that may be used to identify the content item. The content path may include the name of the content item and a folder hierarchy associated with the content item (e.g., the path for storage locally within a client device **102**). In another example, the content path may include a folder or path of folders in which the content item is placed as well as the name of the content item. Content management system **100** may use the content path to present the content items in the appropriate folder hierarchy in a user interface with a traditional hierarchy view. A content pointer that identifies the location of the content item in data store **118** may also be stored with the content identifier. For example, the content pointer may include the exact storage address of the content item in memory. In some embodiments, the content pointer may point to multiple locations, each of which contains a portion of the content item.

In addition to a content path and content pointer, a content item entry/database table row in a content item database entity may also include a user account identifier that identifies the user account that has access to the content item. In some embodiments, multiple user account identifiers may be associated with a single content entry indicating that the content item has shared access by the multiple user accounts.

To share a content item privately, sharing module **126** may be configured to add a user account identifier to the content entry or database table row associated with the content item, thus granting the added user account access to the content item. Sharing module **126** may also be configured to remove user account identifiers from a content entry or database table rows to restrict a user account's access to the content item. The sharing module **126** may also be used to add and remove user account identifiers to a database table for virtual collections.

To share content publicly, sharing module **126** may be configured to generate a custom network address, such as a uniform resource locator (URL), which allows any web browser to access the content in content management system **100** without any authentication. To accomplish this, sharing module **126** may be configured to include content identification data in the generated URL, which may later be used to properly identify and return the requested content item. For example, sharing module **126** may be configured to include the user account identifier and the content path in the generated URL. Upon selection of the URL, the content identification data included in the URL may be transmitted to content management system **100** which may use the received content identification data to identify the appropriate content entry and return the content item associated with the content entry.

To share a virtual collection publicly, sharing module **126** may be configured to generate a custom network address, such as a uniform resource locator (URL), which allows any web browser to access the content in content management system **100** without any authentication. To accomplish this, sharing module **126** may be configured to include collection identification data in the generated URL, which may later be used to properly identify and return the requested content item. For example, sharing module **126** may be configured

to include the user account identifier and the collection identifier in the generated URL. Upon selection of the URL, the content identification data included in the URL may be transmitted to content management system 100 which may use the received content identification data to identify the appropriate content entry or database row and return the content item associated with the content entry or database TOW.

In addition to generating the URL, sharing module 126 may also be configured to record that a URL to the content item has been created. In some embodiments, the content entry associated with a content item may include a URL flag indicating whether a URL to the content item has been created. For example, the URL flag may be a Boolean value initially set to 0 or false to indicate that a URL to the content item has not been created. Sharing module 126 may be configured to change the value of the flag to 1 or true after generating a URL to the content item.

In some embodiments, sharing module 126 may also be configured to deactivate a generated URL. For example, each content entry may also include a URL active flag indicating whether the content should be returned in response to a request from the generated URL. For example, sharing module 126 may be configured to only return a content item requested by a generated link if the URL active flag is set to 1 or true. Changing the value of the URL active flag or Boolean value may easily restrict access to a content item or a collection for which a URL has been generated. This allows a user to restrict access to the shared content item without having to move the content item or delete the generated URL. Likewise, sharing module 126 may reactivate the URL by again changing the value of the URL active flag to 1 or true. A user may thus easily restore access to the content item without the need to generate a new URL.

Sample User Interfaces

Next described are exemplary user interfaces, and various exemplary process flows, which may be implemented in an exemplary system such as is depicted in FIG. 1, to provide a content item purging functionality.

FIGS. 2 through 6 next described illustrate exemplary screenshots that may be provided to a user in accordance with exemplary embodiments of the present invention.

These screenshots implement the goal of some of the embodiments of the present invention which is to provide a way to free space on a user device by automatically deleting content items, such as photos and videos, which have already been uploaded to a content management system, for example.

The following scenarios represent exemplary use cases according to various exemplary embodiments of the present invention.

Scenario 1: A user goes hiking, returns home and runs a photo uploading program such as, for example, camera upload on his user device. The next day the same user returns to the trail and his user device is not connected to any network. After taking a few pictures the camera is nearly full and it prompts the user to delete pictures from a previous day.

Scenario 2: The user has owned a smartphone for a year. It's full of pictures and she is now low on space. She takes another picture and is prompted to clean up the pictures stored on the smartphone that are older than 30 days.

Scenario 3: A user owns a relatively low-end, small device. After taking 10 pictures it is full. An exemplary content management system application running on the device offers to delete them for the user once they have been uploaded.

In exemplary embodiments of the present invention any photo or any video that was camera uploaded (or, for example, other content types, such as audio recordings, MP3s, etc.) may be purged from the user device. If a useful amount of space may be reclaimed by such a purge, an exemplary application may display a notification to a user. This is shown for example in FIG. 2. With reference thereto, an exemplary application entitled "Camera Roll Housekeeping" notifies the user that the application has found 55 photos and 3 videos that were already uploaded to a content management system. It is noted that "Camera Roll" is a common application for storing photos on a smartphone, and thus, for illustrative purposes, the exemplary content purging application of FIGS. 2-6 is similarly named.

For ease of illustration, the generic term ServiceSystem is utilized in FIGS. 2 through 6 to represent a generic content management system, and sometimes the actual datastore in that system is referred to generically as the CMS Datastore. Continuing with reference to FIG. 2, the notification asks the user if she would like to delete those files from her device and free up 185 megabytes of storage space. Finally, the notification provides two active buttons for the user: one stating "No Thanks" and the other stating "Delete". If the user presses delete the purge may begin. If "No Thanks" is chosen, logic may be implemented such that if after K such prompts, all of which result in a choice of "No Thanks" by the user, then no further prompts are sent, or none are sent for a significant time period. Such a user may simply not be interested in automated or assisted purging, and need not be bothered.

In general, not every photo or video that has already been uploaded to an exemplary content management system needs to trigger prompting a user with a notification such as that illustrated in FIG. 2. In fact, as may be seen in FIG. 2, there needs to be a minimum useful amount of space that may be reclaimed or it may not be worth occupying the user's time with such notifications. Exemplary logic may therefore be, for example, as follows:

When DEVICE FREE SPACE is below X percent AND a content item purge will free up more than Y percent of storage (where $Y \gg X - Z$, and Z is the current free space on the device),

an exemplary application or functionality may offer to delete all pictures and/or videos that (i) have been previously uploaded to the content management system and that are (ii) older than, for example, 30 days. In addition, in some embodiments the 30-day (or other "oldness" threshold, as may be chosen) restriction may be broken (but not the "already uploaded" restriction) as much as is needed to meet a minimum amount of cleared space. Using such exemplary logic an exemplary application may only offer the content item purging feature via notification to user, for example, where (i) the user is really low on space, is (ii) taking a picture, and (iii) a meaningful amount of space may be freed up by doing the purge. In general, only material that is older than N days need be purged unless that is not sufficient to free up the space for the user. If it is insufficient, then the application may instead offer to delete whatever is needed in order to meet a sufficient amount of free space, as defined by a given application or system parameter. In such case an application would seek to delete content items from oldest to newest, until a sufficient amount of space is reclaimed.

In exemplary embodiments of the present invention the scan may be optimized so as not to negatively impact battery or system performance. This may be done, for example, by

11

checking battery charge each time, and deciding how aggressively to scan, using, for example, the following logic:

```
IF plugged in to power→scan;
ELSE IF battery charge above threshold J %→scan;
ELSE IF battery charge below threshold K %→do not scan;
```

and using a sliding scale (e.g., only scan upon the occurrence of defined criteria) for battery charge between K and J percentages. K may be set, for example, at 20, 15, 10 percent, or the like, as may be appropriate, depending upon the power draw for deleting content items on that device. Similarly, J may be set, for example at 50-70 percent. In this context it is noted that in many devices scanning and deleting a photograph or other content item may be a fraction of the power needed to take a photograph, for example, and thus not really a concern. Deleting is generally rather inexpensive battery/CPU/disk wise in most cases, inasmuch as in most systems a delete does not actually scrub the actual bits from storage or disk. There is a nuance on some devices in that deletes will necessitate a media scanner to re-index that file, but that is also very inexpensive in terms of power usage.

In exemplary embodiments of the present invention there may be various trigger conditions that cause the running of an exemplary content item purge application. For example, in some embodiments, whenever a user takes a photo or a video the application may check to see if space may be reclaimed. Alternatively, a trigger condition may be any time device free space falls below X percent of the total, and a potential purge would free-up at least Y percent of space. After such checking the user may be notified, as provided in FIG. 2, if the device free space is below X percent and more than Y percent of space may be freed up by the purge. If it is determined that a “useful” amount of space may in fact be recovered by purging photos, videos or other content items, in exemplary embodiments of the present invention such a notification may be issued alerting the user that N megabytes of space may be saved. Such an exemplary notification is that depicted in FIG. 2.

FIG. 3 presents an alternate user interface to that of FIG. 2 with a slight change in the active buttons. In this case the active buttons are “Not Now” and “Yes”. With reference to FIG. 3, if the user clicks “Yes”, or similarly with reference to FIG. 2 if the user clicks “Delete”, a confirmation interface may be displayed such as is shown in FIG. 4. The notification of FIG. 4 repeats the number of photos and/or videos that were already uploaded to a content management system and repeats the question to the user as to whether he or she wishes to in fact delete them, for confirmation. If the user clicks the “Delete” active button then they may be deleted and the space saved or reclaimed. If the user presses “Cancel” then the notification of FIG. 2 or 3 may not be displayed again until the next triggering condition (e.g., taking some video).

Continuing with reference to FIG. 2, in exemplary embodiments of the present invention if the user clicks “No Thanks”, an exemplary system need not again suggest to reclaim storage space until this space is freed (e.g., by the user on her own) and then once again filled up. At such time, if the user once again clicks “No Thanks”, in some embodiments the prompt never need be provided to the user again. Alternatively, in some embodiments a number of days may pass and the prompt may return in case the user changes their mind or decides they may wish to utilize the application.

12

In exemplary embodiments of the present invention, a separate notification to the user may be provided in the event that although some of the photos and/or videos were previously uploaded from the user device to a content management system or other data store they are no longer there, i.e. they have been deleted from the data store of the content management system such that they are no longer available except perhaps on one or more local user devices. In this instance one may assume that the user may have intentionally decided that he or she no longer needs them and that’s why they were deleted from the data store and the copy that is still stored locally is a remnant. Alternatively, the user may decide that she only needs the copy on the local smartphone, for example, and does not wish to delete that “last copy” for the time being. This is the situation depicted in FIG. 5. FIG. 5 is essentially analogous to the confirmation interface of FIG. 4 which is displayed to a user who has clicked the “Delete” button in FIG. 2 or alternatively the “Yes” button in response to the notification of FIG. 3. In FIG. 5 the user is reminded that of the, for example, 55 videos and photos and 3 videos that may be deleted, 2 of the photos are no longer in the data store and so the user should ensure he or she no longer needs them before proceeding. In similar fashion to FIG. 4, there are two possible responses displayed as active buttons, either “Cancel” or “Delete”. If the user presses “Delete” all of the photos and videos including the two that are no longer in the data store will be deleted. In alternate embodiments, for example, a user may be offered the choice to keep the “last copy” items and delete the rest, or, for example, to first view a thumbnail of the “last copy items” prior to choosing a course of action.

Thus, FIGS. 3-5 present a collection of user interface flows that may be provided to a user in some embodiments of the present invention. These are presented synoptically in FIG. 6. Beginning at the top of FIG. 6, with screenshot 610 we see the familiar FIG. 3 notification to the user that a number of photos and videos that were already uploaded to a content management system were located. The user is further notified that should she like to delete these files from the device 185 MB of space would be freed up. The user is provided with two active buttons, as noted above, “Not Now” and “Yes”. If the user activates the Not Now button nothing further happens and process flow ends. Should the user, however, activate the Yes button process flow moves to 620 where the user is displayed the confirmation user interface or notification. There, at the top of the exemplary display the numbers of photos and videos that will be deleted are repeated, and the interface seeks confirmation to delete from the user. Again, there are two active buttons by which the user may signal his or her choice, “Cancel” and “Delete”. If the user clicks Cancel nothing further happens and process flow ends, and in this instance the photos and videos are not deleted. If, on the other hand, the user activates the Delete button the photos and videos as indicated are in fact deleted, the space is reclaimed and process flow ends. Alternatively, if at 610 the user activates the Yes button and, as noted above, some of the photos and/or the videos which were located by the exemplary application are no longer in the data store of the user’s content management system then a warning notification is included in the confirmation user interface 630. 630 is identical to 620 except for the following legend: “2 of the photos you’re about to delete are no longer in your [CMS Datastore]. Ensure you no longer need them before proceeding.”

In similar fashion to the options available in 620 and in FIG. 4, there is “Cancel” and “Delete.” If the user activates the cancel button no deletions occur and process flow ends.

13

If, on the other hand, the user activates the Delete button at 630 then the 55 photos and 3 videos are in fact deleted, even though some of them are no longer in the data store of the content management system. Thus, the user loses the local copies of those photos and/or videos but at the same time 5 reclaims the space as represented to him or her in the initial notification at 630.

Exemplary Process Flows

Next described with reference to FIGS. 7-9 are exemplary process flows according to exemplary embodiments of the present invention. As seen above with reference to FIGS. 2-6, there are various thresholds and variables that may be defined by a system, by a user, or by a combination of both, in exemplary embodiments of the present invention. These parameters may control, for example, when a user is advised of a potential content item purge, and what the notification may look like. Thus, any process flow presented below is solely exemplary given that further conditions may be added or removed as may be desired in various contexts. Having noted that, FIG. 7 depicts a basic process flow according to some embodiments. FIG. 8 is a more detailed process flow, with various conditions controlling the flow. Finally, FIG. 9 is even more detailed, and illustrates the various possible conditions and parameterizations possible. A variety of other process flows similar to that depicted in FIG. 9 are also possible upon instantiation of specific conditions, as noted below.

With reference to FIG. 7, process flow begins at start 700. From start, process flow moves to 710 where an exemplary system may identify content items stored on a user device that have been previously uploaded to a data store. Such a data store may be, for example, a content management system. Such exemplary content items may be, for example, photos, files, audio files such as mp3, or the like. Common to all these examples is the fact that a user's device, for example a smart phone, has a fixed memory capacity, as described above, and additional content items cannot be added to that memory once it reaches capacity. A client process running on the user's device may determine if the user has uploaded to his or her account one or more content items. Such a process may be, for example, a client of a content management system.

Continuing with reference to FIG. 7, process flow then moves to 720 where a determination may be made as to whether the content items identified at 710 are, in the aggregate, sufficiently many in number or in size so as to occupy at least a defined threshold of storage space. If the aggregate of the items identified by the exemplary process of FIG. 7 are in excess of such a defined threshold, then it pays to alert the user that significant storage space on the user device may be reclaimed. If the aggregate of said content items is miniscule, it is not worth taking up the user's time with a process that will effectively not yield much benefit. The defined threshold in exemplary embodiments of the present invention may be expressed as, for example, a fixed number of megabytes of storage, such as for example 100 MB or 500 MB, or, for example, the threshold may be a percentage of the overall storage space located on the device. Alternatively, the threshold may be dynamic, and be lowered the closer a user comes to being at full memory capacity. From 720 process flow moves to 730, where the identified content items may be deleted from the user device. In this exemplary process flow there is no condition imposed of presenting the option to the user and seeking the user's consent. This is a basic process flow and in various exemplary embodiments may be pre-programmed by a user to periodically identify when a significant set of content items

14

is redundant both on the local device and on a content management system and simply clean up the space without even bothering the user. In other exemplary embodiments, such as those whose process flow is depicted in FIG. 8, as well as those depicted above with reference to FIGS. 2-6, a user may always be prompted as to whether or not he or she desires to have the identified content items deleted and space reclaimed, and, for example, if the user signals consent, a further prompt may be sent to the user, asking that he or she confirm the deletion. The prompt may list the aggregate number of content items and the corresponding space savings, such as is shown in FIG. 3, or the prompts may be more detailed, especially as in the case of FIG. 5, where alternatively the photos that are "last copies" may be identified in some way (e.g., thumbs).

From 730 process flow moves to 740, where the user is notified about the deletion and the amount of reclaimed storage as a result. This is an optional step in the context of the process flow of FIG. 7 where, as noted above, a user may simply desire for this processing to happen in the background and need not be advised when sufficient storage has been reclaimed. However, in other exemplary embodiments it may be useful to be made aware when a significant bulk of memory has been reclaimed, especially if the user has a particular use in mind such as, for example, recording lectures or conversations, or taking a large number of photos. From 740 process flow moves to 750, where it ends.

FIG. 8 depicts an alternate exemplary process flow according to some embodiments of the present invention. With reference thereto, process flows begins at start 800, and then moves to 810 where a user device may be monitored for a triggering event to launch a content item purge functionality. It is noted that this process preferably runs in the background on a user device, and thus does not require the user to actually open a client on the user device, such as, for example, a content management system application. In contexts when access to a user device is less readily available, exemplary content item purging can run whenever the client software is open on the device, and/or triggered when a user interacts with content items via the client software, for example.

Continuing with reference to 810, a triggering event may be, for example, a regularly scheduled task, such as checking for potential purges every three weeks, or every two weeks, depending upon the rate at which a user fills the memory on their local device. This parameter may be set by a user, by a system, or by some combination of both, such as, for example, a default system setting but one which a user may alter depending upon their use of memory. Or, for example, based on user usage of content item acquisition (e.g., taking photographs, or recording audio) and memory capacity available, be set dynamically. Alternatively, a triggering event may be the user taking photographs, or invoking a recording functionality on the device such as, for example, may be done by a university student who regularly records her lectures. Doing that with a number of classes may quickly eat up lots of memory, especially if a high resolution or lossless format is used (e.g. ".wav") and the classes are an hour or more in length. As another example, the memory on the user's device having reached some percentage of capacity may be defined as a triggering event. The precise percentage may be, for example, 90%, or 95%, and again may be set as a default, subject to modification by the user, or may be allowed to vary dynamically.

If a potential triggering event is detected at 810, process flow may move to 820 where it is queried as to whether the

15

triggering event has occurred. If no, process flow returns to **810** and the device continues to monitor for a triggering event.

If a triggering event has in fact occurred as determined at **820**, process flow moves to **830** where, similarly to the process of FIG. 7, content items may be identified that have been (1) stored on the user device for more than N days, and also (2) have been previously uploaded to a data store. This is an example of a further condition on the basic process flow of FIG. 7, where it is not enough to simply identify content items that have already been uploaded to a data store, and thus the locally stored items are duplicate content items for that user, but rather that the duplicates are not seen to be worth removing unless they are in some sense “old” and thus not necessary to have on the user device. In some embodiments, “old” may be defined by a system parameter, or by a user, or by a combination of both, to be some number of days. For example, 30 days may be considered to be “old” in the realm of a user who frequently takes pictures and has been known to fill her memory on multiple occasions. Or, for other people, N days may be set at 7 days, as in the case of the college student uploading multiple recordings of lectures she attends.

It is noted that the “old” parameter may be dynamic, and a function of other parameters, such as, for example, memory capacity used. If memory capacity is at, say 96%, what is considered “old” may significantly change, as it is necessary to have some memory space cleared rather urgently. In some embodiments intelligence may be provided to dynamically set the N days of oldness parameter as a function of both memory capacity used and total memory potentially reclaimed. Thus, N may be chosen so as to be small enough to achieve a significant freeing up of memory, subject to a minimum of say, 5 days, for example.

Returning to FIG. 8, Given that a number of content items have been identified at **830** to have met the condition of being “old” by simply being on the user device for at least N days—as that parameter is then dynamically defined—process flow may move to **840**, where a test is performed to determine whether sufficient amount of memory is occupied by these items to justify a purge. This is similar to the test of FIG. 7 at **720** where it is queried whether in the aggregate the content items identified for potential deletion occupy at least a defined threshold of storage space. Here the storage space threshold is expressed in number of megabytes X occupied by the identified content items, which again may be a system parameter. Alternatively, instead of defining the threshold in terms of megabytes of memory, it could be a percentage of overall memory capacity, as noted above in the discussion of FIGS. 2-6.

As noted, a triggering event such as indicated at **810** and **820** may also be the user device being within Y percent of the overall memory capacity. For example, if the user has filled 95% or more of the available storage space, the user is really in a critical condition and a purge of content should be performed. Otherwise she will quickly run out of memory. When a user device is close to being full and its memory saturated, the notion of age or “old” may become quite short in terms of days N, and the minimum threshold of X megabytes that may be reclaimed may also become quite small. Alternatively, if the triggering event is not the user device being near full capacity of memory, but rather other criteria such as taking a photo or a periodically scheduled purge offer, the parameters N days of aging and X megabytes of potentially reclaimed memory may be set longer and higher, respectively, as the case may be. In a

16

general sense, all of these parameters may be understood as “capacity and space conditions”, as described below with reference to FIG. 9.

Returning now to **840**, if the content items identified do in fact occupy greater than X megabytes of memory, then process flow moves to **850** for further testing. If they do not, then process flow may move back to **810** where the user device is once again monitored for a triggering event. Continuing with **850**, where the items do occupy sufficient storage space to make a purge worthwhile, it may be determined whether any of the content items are no longer in the user’s datastore at **850**. This is important because although they were once uploaded to her datastore, the items on the local user device may now be the last copy available to this user given that she may have deleted those uploaded items from the datastore at some time subsequent to uploading them. If this is the case, either the user does not care about them—in which case she may not mind at all removing them from the local device—or she may have decided to “clean house” at the datastore in reliance upon having a local copy on her smart phone, for example. Therefore, process flow will bifurcate at **850** depending upon the answer to the determination. If yes at **850**, then process flow moves to **860** where the user is informed of possible deletion of content items and space to be reclaimed as result, in a similar notification to that of FIG. 5. As well, the user may be advised as to which items in particular on her device are “last copy” items, i.e., items which were once uploaded to the datastore but are no longer present in the datastore, and asked if she is interested deleting some or all of the identified content items. In general, a user will, if interested in purging memory space, freely delete items which are on the datastore and may delete items which are no longer in the datastore. If at **860** the user confirms a desire to delete some or all of the available content items, then process flow may move to **870** where the exemplary system confirms that the user is interested in the deletion. Alternatively, and apposite to FIG. 5, following an initial query such as that of FIG. 2 or 3, it may be only first brought to the user’s attention in the delete confirmation notification that some of the content items are “last copies”.

Returning now to **850**, where, if the answer to the determination is no, that means that no items tagged for potential deletion are actually last copies, then the user is informed of the possible deletion and the space that may be reclaimed as a result and is asked, as is shown in FIG. 2 if he or she is interested in deleting those items at **865**. Process flow moves to **870** where it is determined if (1) the user is in fact interested and (2) whether the user confirms the deletion (as seen in FIGS. 3 and 4, respectively), and if she responds yes to both, process flow moves to **875** where the content items authorized by the user are deleted from the user device. From there process flow may move to **880** where the user is notified of the deletion and the amount of reclaimed storage, and process flow ends at **890**.

FIG. 9 is a similar process flow to that of FIG. 8, with some important exceptions. In similar fashion as described for FIG. 8, process flow begins at **900** and determines at **920** if a triggering event for the content item purge process has occurred. If no, process flow returns to **910**. However if yes, then at **930** content items are identified that (1) were previously uploaded, and (2) are “old or otherwise removable.” this latter condition is an extension of the “oldness” idea of FIG. 8, where besides merely being old, an item may be designated as removable for various other reasons. This may include a better copy being already on the datastore, albeit not one uploaded from the smartphone, for example.

Or, for example, a better copy being shared with the user on the CMS, and thus fully available to the user. Such considerations may even break the “previously uploaded” condition in some embodiments. Moreover, the removability conditions are dynamic, as next noted.

At 940 the current capacity of the memory (justifying a purge) and the size of memory occupied by the identified items may be analyzed. If the latter is small, but capacity is near full, although the condition is not satisfied, flow returns to 910 and removability conditions are informed at step 945, so that they may be dynamically modified. Such modification may make more content items (usually newer ones) eligible for removal, and thus the next time the process runs (which may be immediately, if being near memory capacity is a triggering event) more items may be designated as removable. This illustrates dynamically modifying items eligible for removal in response to then current memory conditions, which may be implemented in some embodiments.

Flow through 940, 950, 960 and 965, 970, 975, and 980 are essentially identical to that of the analogous steps of FIG. 8, and thus will not be repeated. However at 970, if the user does not confirm the deletion, instead of process flow returning to 910, a test may be performed at 985 as to whether to continue. This covers the situation where a user has repeatedly declined to perform any deletions. If so, logic may preclude further prompts of the content item purging service, and process ends at 990.

General Considerations and Extensions

In general, embodiments of the invention allow a user to keep content items on a device for a minimum amount of time and if you there is ample memory space on the device there is no concern to delete anything. However, if the user took a whole of bunch of pictures yesterday, the idea is to free the space after they have been removed in the event the device is running out of space. If the user took a bunch of pictures from say two months ago but has plenty of space, there is no need to even ask the user.

In some embodiments, a triggering event may be simply downloading of a large file or files in general, so that if a user downloads such a big file, space may be freed up by purging content items. Thus, for example, the user may download things to a download folder and they may end up with a lot of junk in there. An exemplary application may transfer that stuff to a remote datastore, have it available, saved and bookmarked and then delete the downloads folder to reclaim space.

As noted, the capacity threshold may be determined by the device’s free space. When low on space, say less than 10% free space, in some embodiments there may be a sliding window where content items may be aggressively deleted that are older than say 12 hours, even. Otherwise, in other embodiments, delete photos older than say 30 days. If a useful amount of space of roughly 1% of the device capacity may be claimed, a notification may be displayed to the user.

Additionally, in alternate embodiments, content items need not be explicitly stored on the content management system from the user device. Rather, content items may be purged from the user device as long as a copy exists on the content management system, or even, in some embodiments, if a similar image or content item is stored on the CMS. Such similarity may be determined using criteria, methods and embodiments as are disclosed in U.S. Utility patent application Ser. No. 13/888,082, filed on May 6, 2013, entitled “Duplicate/Near Duplicate Detection and Image Registration”, the disclosure of which is hereby incorporated by reference in its entirety.

As an interesting extension, in some embodiments as a user takes photographs, for example, when they are uploaded to a CMS they may be replaced on the local user device with a downsampled, or truncated version. So there may be a different view than the native view locally, and the native view would be replaced with smaller copies of the same photos. This is not a true purge, but a “shrinkage” and may be used to save space if removability conditions do not allow a purge, for example. Thus, for example, this reduces the amount of space that was taken on the phone by the content item, instead of entirely getting rid of it. In some embodiments a default setting may be to do this with every photograph taken, thus obviating the need for true purges.

Thus, in such alternate embodiments, when a user takes pictures, the exemplary application would store the full size until it may be moved to the CMS servers. Subsequently, when the user opens the photo, she is shown a high quality preview and if she then really wants to, she may retrieve the full size on the phone, for example.

In some embodiments a triggering event may be a request by a user. One way this may be done is the user goes into his mailbox after being prompted inside an exemplary application.

In a related extension, attachments to emails may also be purged to reclaim space. Thus, for example, a user’s sibling sends family photos that taking up a bunch of space in the user’s email account, and the user needs to free up space within the email client. They may be uploaded to a CMS and then removed from the user’s email account so they are not taking up space. Effectively email archiving via a content management system.

Thus, in general, by leveraging an exemplary local application’s ability to upload to a content management system, various content items of a wide variety of types may be archived on the content management system, and either purged, or stored in a much smaller “preview” “thumb” or otherwise truncated form on the user device.

Exemplary Implementations

Any suitable programming language may be used to implement the routines of particular embodiments including, but not limited to, the following: C, C++, Java, JavaScript, Python, Ruby, CoffeeScript, assembly language, etc. Different programming techniques may be employed such as procedural or object oriented. The routines may execute on a single processing device or multiple processors. Although the steps, operations, or computations may be presented in a specific order, this order may be changed in different particular embodiments. In some particular embodiments, multiple steps shown as sequential in this specification may be performed at the same time.

Particular embodiments may be implemented in a computer-readable storage device or non-transitory computer readable medium for use by or in connection with the instruction execution system, apparatus, system, or device. Particular embodiments may be implemented in the form of control logic in software or hardware or a combination of both. The control logic, when executed by one or more processors, may be operable to perform that which is described in particular embodiments.

Particular embodiments may be implemented by using a programmed general purpose digital computer, by using application specific integrated circuits, programmable logic devices, field programmable gate arrays, optical, chemical, biological, quantum or nanoengineered systems, components and mechanisms may be used. In general, the functions of particular embodiments may be achieved by any means as is known in the art. Distributed, networked sys-

19

tems, components, and/or circuits may be used. Communication, or transfer, of data may be wired, wireless, or by any other means.

It will also be appreciated that one or more of the elements depicted in the drawings/Figs. may also be implemented in a more separated or integrated manner, or even removed or rendered as inoperable in certain cases, as is useful in accordance with a particular application. It is also within the spirit and scope to implement a program or code that may be stored in a machine-readable medium, such as a storage device, to permit a computer to perform any of the methods described above.

As used in the description herein and throughout the claims that follow, “a”, “an”, and “the” includes plural references unless the context clearly dictates otherwise. Also, as used in the description herein and throughout the claims that follow, the meaning of “in” includes “in” and “on” unless the context clearly dictates otherwise.

While there have been described methods for contact importing and processing, it is to be understood that many changes may be made therein without departing from the spirit and scope of the invention. Insubstantial changes from the claimed subject matter as viewed by a person with ordinary skill in the art, no known or later devised, are expressly contemplated as being equivalently within the scope of the claims. Therefore, obvious substitutions now or later known to one with ordinary skill in the art are defined to be within the scope of the defined elements. The described embodiments of the invention are presented for the purpose of illustration and not of limitation.

What is claimed is:

1. A method of reclaiming storage space on a user device, comprising:

detecting a triggering event for reclaiming storage space on the user device;

determining an amount of storage space on the user device to be reclaimed; and

in response to detecting the triggering event:

identifying one or more content items stored on the user device that qualify for content item purging;

wherein a content item stored on the user device qualifies for content item purging only if the content item is previously synchronized with a remote data store and is not modified at the user device since the content item was previously synchronized with the remote data store;

wherein identifying the one or more content items that qualify for content item purging comprises determining that each of the one or more identified content items is previously synchronized with the remote data store and is not modified at the user device since the content item was previously synchronized with the remote data store;

determining whether deleting the one or more identified content items from the user device would reclaim at least the amount of storage space on the user device; deleting the one or more identified content items from the user device;

wherein deleting the one or more identified content items from the user device reclaims at least the amount of storage space on the user device; and wherein the method is performed by an electronic device.

2. The method of claim 1, wherein previous synchronizing causing at least a portion of at least one of the one or more content items to be stored in the remote data store.

20

3. The method of claim 1, wherein said triggering event is at least one of:

a periodically scheduled purge,

a user request, or

a receipt of an additional content item.

4. The method of claim 1, wherein detecting the triggering event comprises determining that available space in memory on the user device is less than a first memory threshold value.

5. The method of claim 1, wherein the one or more content items are associated with a user account of a content management service held by a user of the user device.

6. The method of claim 1, wherein the identifying step comprises identifying one or more content items stored on the user device that have been stored on the user device for at least a defined time period and that have been previously synchronized with the remote data store.

7. The method of claim 6, wherein the defined time period is one of: fixed, set as a default but modifiable by a user, or dynamically set as a function of memory capacity and/or content type.

8. The method of claim 1, wherein the one or more content items stored on the user device that have been previously synchronized with the remote data store are versions of one or more corresponding synchronized content items stored in the remote data store.

9. The method of claim 4, wherein the first memory threshold value comprises a first predefined percentage of a total capacity of the memory of the user device.

10. The method of claim 1, wherein detecting a triggering event comprises determining that a battery charge of a battery of the user device exceeds a battery charge threshold.

11. The method of claim 1, wherein detecting a triggering event comprises determining that a battery of the user device is currently being charged.

12. The method of claim 1, wherein the identifying step includes determining that an aggregate amount of storage space occupied by the one or more content items on the user device comprises at least a second memory threshold value of the user device's memory.

13. The method of claim 1, further comprising:

in response to detecting the triggering event:

identifying one or more photo or video items stored on the user device that have been previously synchronized with the remote data store; and

replacing the one or more photo or video items with downsampled versions of the one or more photo or video items.

14. The method of claim 1, further comprising:

prompting a user to confirm a content item purge action only if deleting the one or more identified content items from the user device would reclaim at least the minimum amount of storage space on the user device; and deleting the one or more identified content items from the user device only if confirmation to proceed with the content item purge action is received from the user.

15. A non-transitory computer readable medium containing instructions that, when executed by at least one processor of a computing device, cause the computing device to: detect a triggering event for reclaiming storage space on a user device; determine an amount of storage space on the user device to be reclaimed; and

in response to detecting the triggering event:

identify one or more content items stored on the user device that qualify for content item purging;

21

wherein a content item stored on the user device qualifies for content item purging only if the content item is previously synchronized with a remote data store and is not modified at the user device since the content item was previously synchronized with the remote data store;

wherein the instructions that cause the computing device to identify the one or more content items that qualify for content item purging, comprise instructions that cause the computing device to determine that each of the one or more identified content items is previously synchronized with the remote data store and is not modified at the user device since the content item was previously synchronized with the remote data store;

determine whether deleting the one or more identified content items from the user device would reclaim at least the amount of storage space on the user device; delete the one or more identified content items from the user device; and

wherein the instructions that cause the computing device to delete the one or more identified content items from the user device, cause to reclaim at least the amount of storage space on the user device.

16. The non-transitory computer readable medium of claim 15, wherein the instructions that cause the computing device to previously synchronize the one or more content items with the remote data store causes at least a portion of at least one of the one or more content items to be stored in the remote data store.

17. The non-transitory computer readable medium of claim 16, wherein the instructions that cause the computing device to detect the triggering event comprise instructions that, when executed by at least one processor of the computing device, further cause the computing device to detect the triggering event based on determining that available space in memory on the user device is less than a first memory threshold value.

18. The non-transitory computer readable medium of claim 15, wherein said triggering event is at least one of:
a periodically scheduled purge,
a user request, or
a receipt of an additional content item.

19. The non-transitory computer readable medium of claim 15, wherein the instructions that cause the computing device to detect the triggering event comprise instructions that, when executed by at least one processor of the computing device, further cause the computing device to detect the triggering event based on determining that a battery charge of a battery of the user device exceeds a battery charge threshold.

20. The non-transitory computer readable medium of claim 15, wherein the instructions that cause the computing device to detect the triggering event comprise instructions that, when executed by at least one processor of the computing device, further cause the computing device to detect the triggering event based on determining that a battery of the user device is currently being charged.

21. The non-transitory computer readable medium of claim 15, wherein the one or more content items stored on the user device that have been previously synchronized with the remote data store are versions of one or more corresponding synchronized content items stored in the remote data store.

22

22. The non-transitory computer readable medium of claim 15, wherein said one or more items are associated with a user account of a content management service held by a user of the user device.

23. The non-transitory computer readable medium of claim 15, wherein the instructions, when executed by at least one processor of the computing device, further cause the computing device to:

in response to detecting the triggering event:

identify one or more photo or video items stored on the user device that have been previously synchronized with the remote data store; and

replace the one or more photo or video items with downsampled versions of the one or more photo or video items.

24. A system for content item purging on a user device, comprising:

at least one processor;

a display; and

memory containing instructions that, when executed, cause the at least one processor to:

detect a triggering event for reclaiming storage space on the user device;

determine an amount of storage space on the user device to be reclaimed; and

in response to detecting the triggering event:

identify one or more content items stored on the user device that qualify for content item purging;

wherein a content item stored on the user device qualifies for content item purging only if the content item is previously synchronized with a remote data store and is not modified at the user device since the content item was previously synchronized with the remote data store;

wherein the instructions that cause the at least one processor to identify the one or more content items that qualify for content item purging, comprise instructions that cause the at least one processor to determine that each of the one or more identified content items is previously synchronized with the remote data store and is not modified at the user device since the content item was previously synchronized with the remote data store;

determine whether deleting the one or more identified content items from the user device would reclaim at least the amount of storage space on the user device; and

delete the one or more identified content items from the user device; and

wherein the instructions that cause the at least one processor to delete the one or more identified content items from the user device, cause to reclaim at least the amount of storage space on the user device.

25. The system of claim 24, wherein the one or more content items stored on the user device that have been previously synchronized to the remote data store are versions of one or more corresponding synchronized content items stored in the remote data store.

23**24**

26. The system of claim **24**, wherein the instructions further cause the at least one processor to:

in response to detecting the triggering event:

identify one or more photo or video items stored on the user device that have been previously synchronized 5
with the remote data store; and

replace the one or more photo or video items with
downsampled versions of the one or more photo or
video items.

* * * * *

10